COMMONWEALTH SECONDARY SCHOOL
END OF YEAR EXAMINATION 2021

COMPUTING
Paper 2 (Practical)

Name: ____ (   )   Class: ____________

SECONDARY FOUR EXPRESS/5 NORMAL (ACADEMIC) **Wednesday 1 Sept 2021**

7155/2 **0800 – 1030**

2 hour 30 mins

READ THESE INSTRUCTIONS FIRST

Answer **all** questions.

All tasks must be done in the computer laboratory. You are not allowed to bring in or take out any pieces of work or materials on paper or electronic media or in any other form.

Programs are to be written in Python.
Save your work using the file name given in the question as and when necessary.

The number of marks is given in brackets [ ] at the end of each question or part question.
The total number of marks for this paper is 50.

Name of setter: Mrs Leong Sai Hong

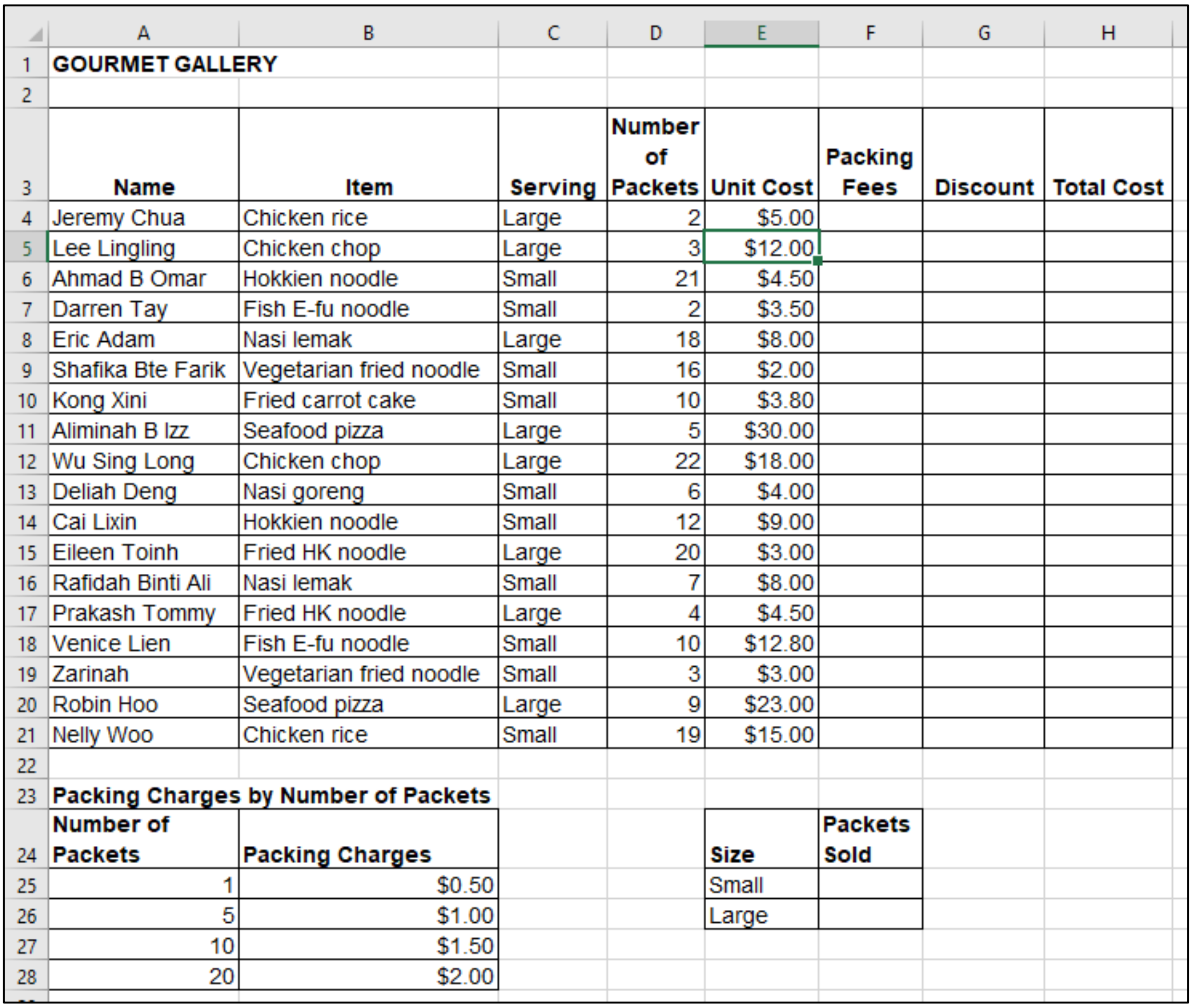
This paper consists of **5** printed pages including the cover page.

[Turn over

Task 1

A food delivery company uses spreadsheet software to record each customer order. Serving sizes comes in “Small” and “Large”. Each customer is supposed to purchase only packets of food of the same serving size. You are required to finish setting up the spreadsheet to calculate the cost each customer has to pay.

Open the file **GOURMET.xlsx** and you will see the following data.



	A	B	C	D	E	F	G	H
1	GOURMET GALLERY							
2								
3	Name	Item	Serving	Number of Packets	Unit Cost	Packing Fees	Discount	Total Cost
4	Jeremy Chua	Chicken rice	Large	2	$5.00			
5	Lee Lingling	Chicken chop	Large	3	$12.00			
6	Ahmad B Omar	Hokkien noodle	Small	21	$4.50			
7	Darren Tay	Fish E-fu noodle	Small	2	$3.50			
8	Eric Adam	Nasi lemak	Large	18	$8.00			
9	Shafika Bte Farik	Vegetarian fried noodle	Small	16	$2.00			
10	Kong Xini	Fried carrot cake	Small	10	$3.80			
11	Aliminah B Izz	Seafood pizza	Large	5	$30.00			
12	Wu Sing Long	Chicken chop	Large	22	$18.00			
13	Deliah Deng	Nasi goreng	Small	6	$4.00			
14	Cai Lixin	Hokkien noodle	Small	12	$9.00			
15	Eileen Toinh	Fried HK noodle	Large	20	$3.00			
16	Rafidah Binti Ali	Nasi lemak	Small	7	$8.00			
17	Prakash Tommy	Fried HK noodle	Large	4	$4.50			
18	Venice Lien	Fish E-fu noodle	Small	10	$12.80			
19	Zarinah	Vegetarian fried noodle	Small	3	$3.00			
20	Robin Hoo	Seafood pizza	Large	9	$23.00			
21	Nelly Woo	Chicken rice	Small	19	$15.00			
22								
23	Packing Charges by Number of Packets							
24	Number of Packets	Packing Charges			Size	Packets Sold		
25	1	$0.50			Small			
26	5	$1.00			Large			
27	10	$1.50						
28	20	$2.00						

Save the file as **MYGOURMET_**<class>_<index_number>_<your name>**.xlsx**

1. In cells **F4** to **F21** enter a formula that uses an appropriate function to search for the **Packing Charges** in the **Packing Charges by Number of Packets** table. Use it to calculate the Packing Fees for each order. (E.g. a customer who orders between 1 and 4 packs will be charged $0.50 for his order.) [2]

2. Use a conditional statement to identify whether a discount is to be given. For customers who order large serving or where the number of packets is greater than 15, put **Yes** in the Discount column. [2]

3. Use a conditional statement to calculate the total cost in the **Total Cost** column. Customer who are identified as **Yes** for the **Discount** must be given a 5% reduction for the total cost. [2]

4. In cell **F25** and **F26**, enter a formula that uses an appropriate function to count the number of small and large packets of food sold. [2]

5. In cell **A4** to **H21**, use a formatting tool to change the colour of the row to yellow if the Unit Cost is greater than $10. [2]

Task 2

The following program allows the user five chances to search for characters within a string.

```
text = "With great power comes great responsibility"

chances = 5

for c in range(chances):
    ch = input("Search character:")
    for t in text:
        if ch == t:
            break
```

Open the file **SEARCH.py**

Save the file as **MYSEARCH_**<class>_<index_number>_<your name>.**py**

6. Edit the program so that:

   (a) For each character that searched, it outputs a suitable message only once at the end of the search if the character is not found in the string. [3]

   (b) It concatenates all characters that are not found in the string and outputs this new string. [3]

7. Save your program as **VARSEARCH_**<class>_<index number>_<your name>**.py**

   Edit your program so that:

   (a) For each character that is searched, it prints out the number of times it is found in the text. [3]

   (b) It works for any number of chances to search for characters. [1]

   Save your program.

Task 3

The Average Subject Grade (ASG) is used to measure a student's overall performance. For each subject, ASG is a number between 1 and 4. The following calculates the ASG using the following rules:

- Convert each subject's score into the grade points using the table below.
- Add up all the grade points.
- Divide the result by the number of subjects.
- Round the result to two decimal places. This will be the student's ASG.
- For example, if a student takes 7 subjects and his scores are 20, 50, 48, 79, 90, 52, 61, his corresponding ASG will be 2.57.

Score	Grade	Grade points
70 - 100	A1	1
60 - 69	B2	2
50 - 59	C3	3
Less than 50	D4	4

There are several syntax errors and logical errors in the program.

```
grade = ['A1','B2','C3','D4']
total=0
num = int(input("Enter number of subjects:"))
score = ['']*num
result=[0]*num

for i in range(num):
    score=int(input("Enter mark:"))

for i in range(len(grade)):
    if score[i]>70:
        result[0]=result[0]+1
    elif score[i]>=60:
        result[1]=result[1]+1
    elif score[i]<=50:
        result[2]=result[2]+1
    else:
        result[3]=result[3]+1

for i in range(len(grade)):
    total=int(grade[i][:1])*result[i]
    print('ASG is',total)
```

Open the file **ASG.py**

Save the file **MYASG_**<class>_<index number>_<your name>**.py**

8. Identify and correct the errors in the program so that it works correctly according to the rules above. [10]

   Save your program.

Task 4

You have been asked to create a guessing game program.

The program should:

- Allow player 1 to input a 4-digit number for player 2 to guess. There must be validation present to check that the input is a 4-digit number. Output a suitable error message if validation test fails.
- Allow player 2 to have 5 chances to correctly guess the number input by player 1.
- When player 2 inputs the same number as player, output the number of guesses and the game ends. E.g. “You guessed correctly in 3 guess(es).” when player 2 guessed the correct number on the 3rd guess.
- Output “You did not guess the correct number.” When player 2 does not input the same number as player 1.
- Output “Game over!” when the player has made 5 incorrect guesses.

9. Write your program and test that it works. [8]
Save your program as **MYSTAR1_**<class>_<index number>_<your name>**.py**

10. When your program is working, use the following test data to show your test result.
- Test 1 – Player 1 inputs abcd
- Test 2 – Player 1 inputs 5
- Test 3 – Player 1 inputs 1234 and player 2 guesses 9934 and 1234

Take a screenshot of your result and save it as
TEST123_<class>_<index number>_<your name> [4]
Save your file in either .jpg or .png format.

11. Save your program as **MYSTAR2_**<class>_<index number>_<your name>**.py**

Extend your program as follows:

- Generate the 4-digit random number where al the digits are different.
  Use the randint() function below:

```
import random
print(random.randint(0,9)) # outputs a random number between 0 and 9
```

- When the player inputs a number that does not match the generated number, output the number of “stars” which are digits found in the generated number in the correct position and the number of “moons” which are the digits found in the generated number but in the wrong position.
  For example, if the generated number is 2678, a guess of 2806 will result in “1 star and 2 moons”. [5]

Save your program.

12. Save your program as **MYSTAR3_**<class>_<index number>_<your name>**.py**
Extend your program to calculate and output the average number of stars at the end of the game. [3]
Save your program.

END OF PAPER